Introduction to Big Data Assignment-5

Name: Faizan Mulla
Roll No: 21F1003885

Problem Statement

Write SparkSQL code to implement SCD Type II on a customer master data frame. You can use the same input files you created for Assignment 4. The SparkSQL code should be executed on a Dataproc cluster.

Solution

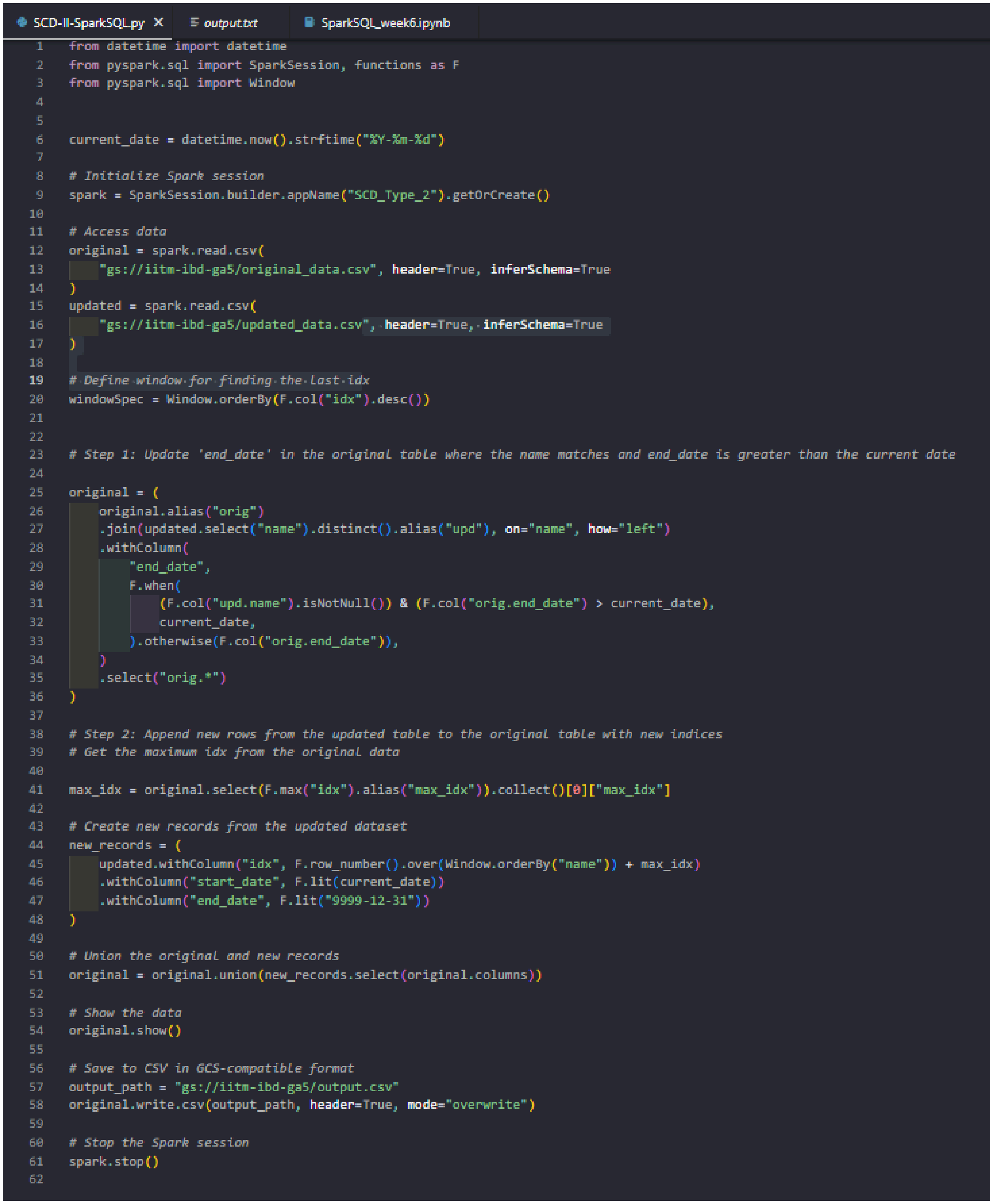
Environment Setup

1. GCP Setup
   - Set up a new project with default service account configurations
   - Enabled necessary APIs (Compute Engine and Dataproc)

2. Dataproc Cluster Creation
   - Created a Dataproc cluster with following specifications:
     - chose the option: Create cluster on compute engine
     - manager node: series → e2 // machine type → e2-standard-2 (2vCPU, 1 core, 8GB)
     - reduce primary disk size from 500GB to something less like 50GB.
     - exact same settings for worker nodes too.
     - Region: **asia-south-1**
     - in the customize cluster menu, **uncheck** the INTERNAL IP ONLY option.

3. Cloud Storage Setup
   - Created a Cloud Storage bucket for storing the original and the updated CSV files + python file
   - Configured with standard storage class
   - Region: **asia-south-1**

Implementation Details



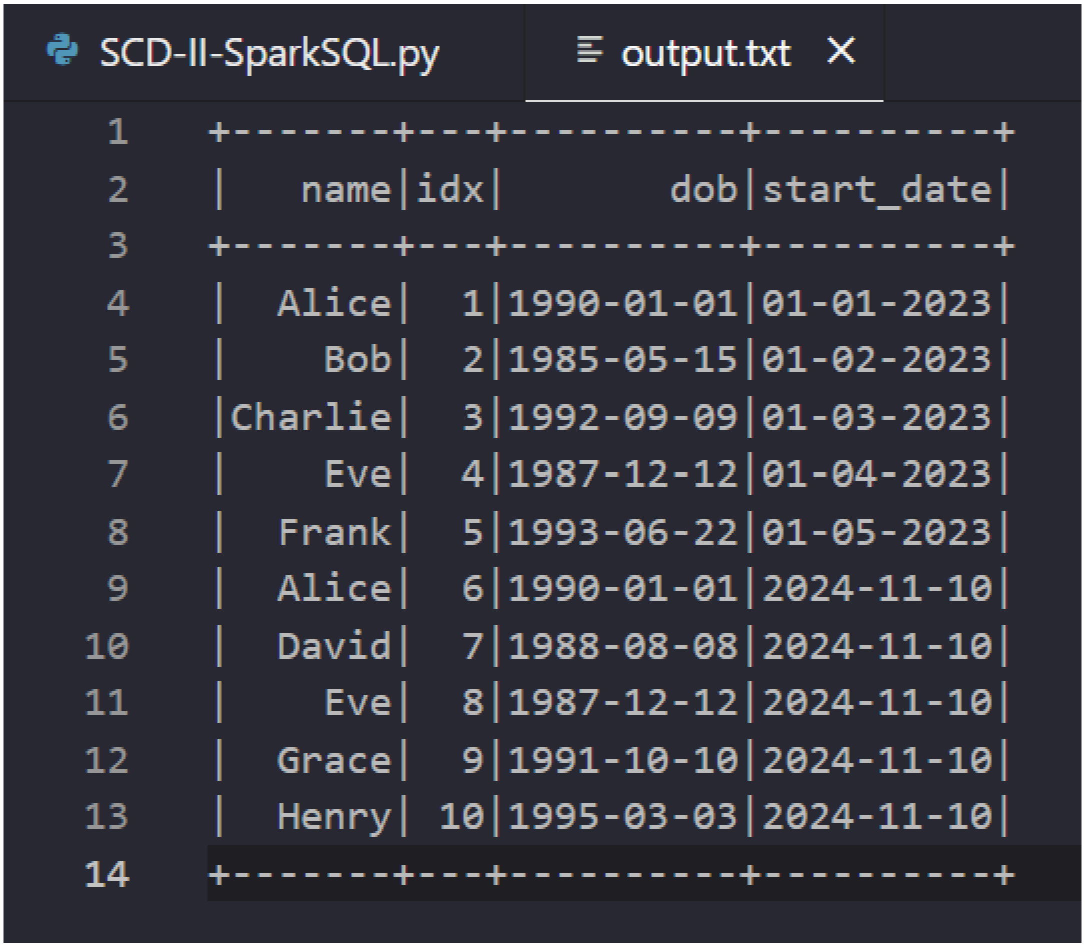
```python
from datetime import datetime
from pyspark.sql import SparkSession, functions as F
from pyspark.sql import Window


current_date = datetime.now().strftime("%Y-%m-%d")

# Initialize Spark session
spark = SparkSession.builder.appName("SCD_Type_2").getOrCreate()

# Access data
original = spark.read.csv(
    "gs://iitm-ibd-ga5/original_data.csv", header=True, inferSchema=True
)
updated = spark.read.csv(
    "gs://iitm-ibd-ga5/updated_data.csv", header=True, inferSchema=True
)

# Define window for finding the last idx
windowSpec = Window.orderBy(F.col("idx").desc())


# Step 1: Update 'end_date' in the original table where the name matches and end_date is greater than the current date

original = (
    original.alias("orig")
    .join(updated.select("name").distinct().alias("upd"), on="name", how="left")
    .withColumn(
        "end_date",
        F.when(
            (F.col("upd.name").isNotNull()) & (F.col("orig.end_date") > current_date),
            current_date,
        ).otherwise(F.col("orig.end_date")),
    )
    .select("orig.*")
)

# Step 2: Append new rows from the updated table to the original table with new indices
# Get the maximum idx from the original data

max_idx = original.select(F.max("idx").alias("max_idx")).collect()[0]["max_idx"]

# Create new records from the updated dataset
new_records = (
    updated.withColumn("idx", F.row_number().over(Window.orderBy("name")) + max_idx)
    .withColumn("start_date", F.lit(current_date))
    .withColumn("end_date", F.lit("9999-12-31"))
)

# Union the original and new records
original = original.union(new_records.select(original.columns))

# Show the data
original.show()

# Save to CSV in GCS-compatible format
output_path = "gs://iitm-ibd-ga5/output.csv"
original.write.csv(output_path, header=True, mode="overwrite")

# Stop the Spark session
spark.stop()
```

Execution Process

1. Data Generation / Uploading
   - Uploaded ***original_data.csv*** & ***updated_data.csv*** files to Cloud Storage Bucket.
   - Changed the file names in the Python script to the gsutil URI of the respective files.

2. SCD-2
   - Now, upload ***SCD-II-SparkSQL.py*** to Cloud Storage
   - Created Dataproc Cluster
   - Submitted Spark job through Dataproc
   - Monitored job execution through Dataproc UI

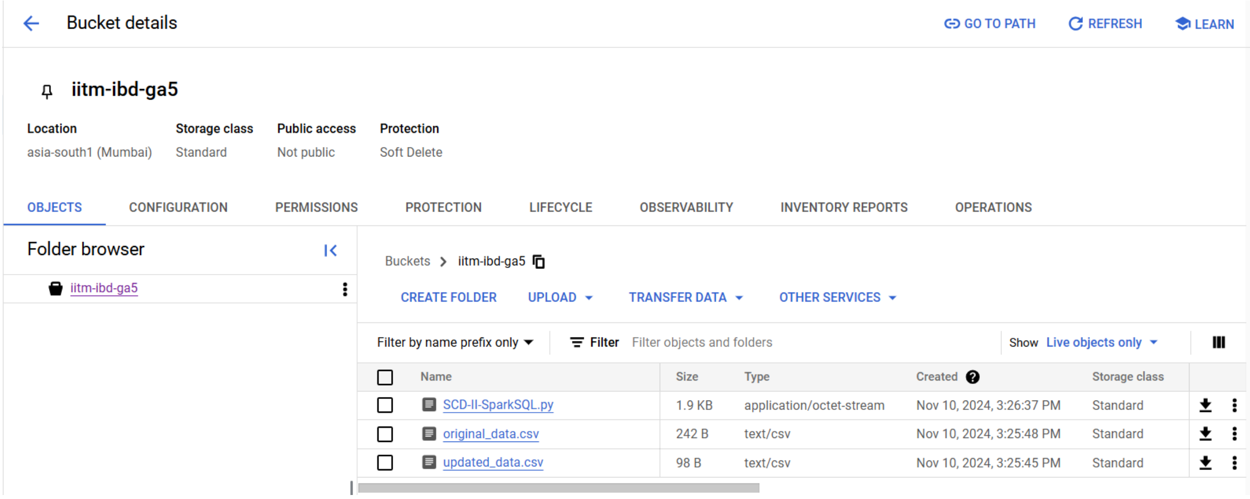
Results



```
+-------+---+----------+----------+
|   name|idx|       dob|start_date|
+-------+---+----------+----------+
|  Alice|  1|1990-01-01|01-01-2023|
|    Bob|  2|1985-05-15|01-02-2023|
|Charlie|  3|1992-09-09|01-03-2023|
|    Eve|  4|1987-12-12|01-04-2023|
|  Frank|  5|1993-06-22|01-05-2023|
|  Alice|  6|1990-01-01|2024-11-10|
|  David|  7|1988-08-08|2024-11-10|
|    Eve|  8|1987-12-12|2024-11-10|
|  Grace|  9|1991-10-10|2024-11-10|
|  Henry| 10|1995-03-03|2024-11-10|
+-------+---+----------+----------+
```

Relevant Screenshots

1. Cloud Storage Bucket Contents



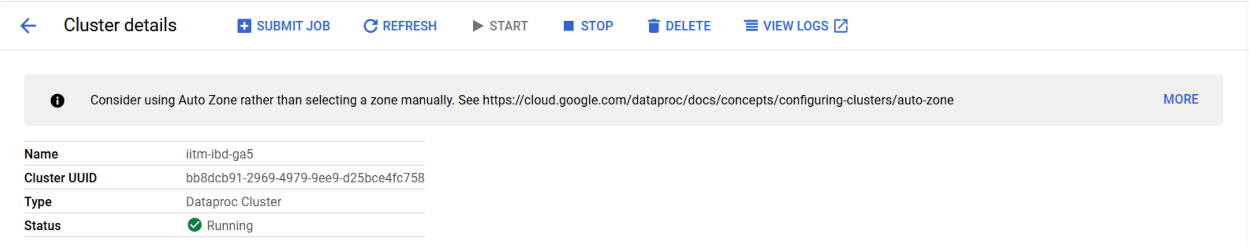
iitm-ibd-ga5

Location	Storage class	Public access	Protection
asia-south1 (Mumbai)	Standard	Not public	Soft Delete

Buckets > iitm-ibd-ga5

Name	Size	Type	Created	Storage class
SCD-II-SparkSQL.py	1.9 KB	application/octet-stream	Nov 10, 2024, 3:26:37 PM	Standard
original_data.csv	242 B	text/csv	Nov 10, 2024, 3:25:48 PM	Standard
updated_data.csv	98 B	text/csv	Nov 10, 2024, 3:25:45 PM	Standard

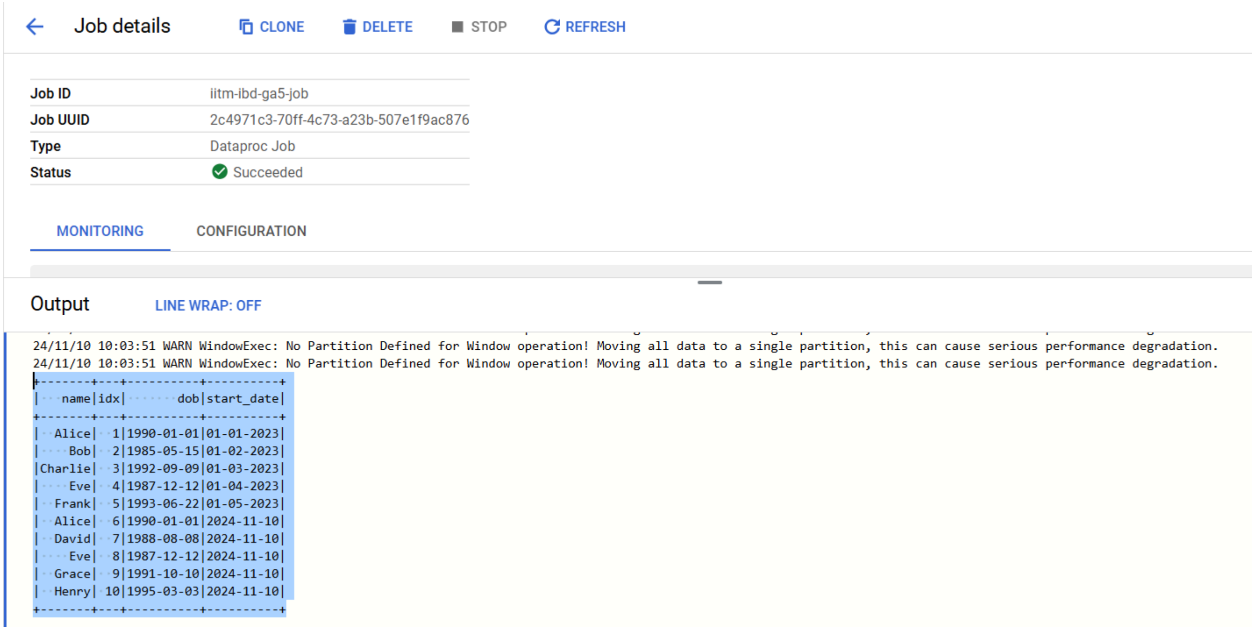
2. Dataproc Cluster Configuration



Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone

Name	iitm-ibd-ga5
Cluster UUID	bb8dcb91-2969-4979-9ee9-d25bce4fc758
Type	Dataproc Cluster
Status	✓ Running

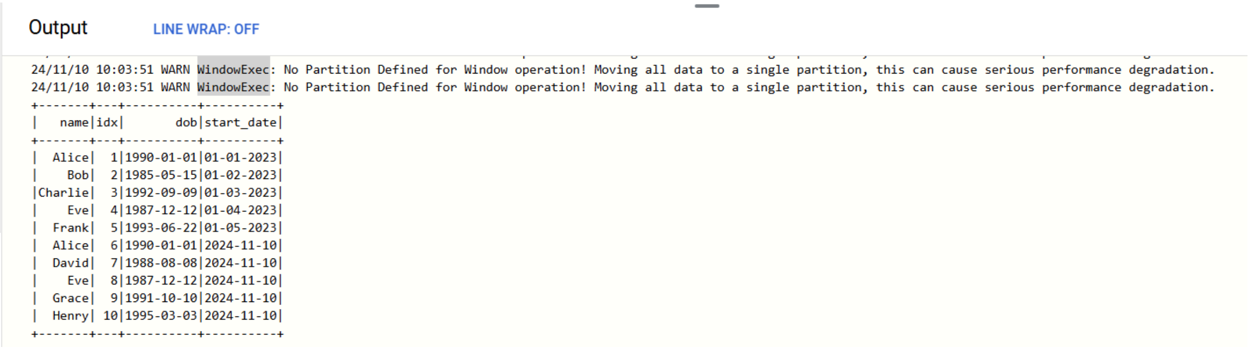
3. **Job Execution Results**



Job ID	iitm-ibd-ga5-job
Job UUID	2c4971c3-70ff-4c73-a23b-507e1f9ac876
Type	Dataproc Job
Status	✓ Succeeded

```
24/11/10 10:03:51 WARN WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
24/11/10 10:03:51 WARN WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
+-------+---+----------+----------+
|   name|idx|       dob|start_date|
+-------+---+----------+----------+
|  Alice|  1|1990-01-01|01-01-2023|
|    Bob|  2|1985-05-15|01-02-2023|
|Charlie|  3|1992-09-09|01-03-2023|
|    Eve|  4|1987-12-12|01-04-2023|
|  Frank|  5|1993-06-22|01-05-2023|
|  Alice|  6|1990-01-01|2024-11-10|
|  David|  7|1988-08-08|2024-11-10|
|    Eve|  8|1987-12-12|2024-11-10|
|  Grace|  9|1991-10-10|2024-11-10|
|  Henry| 10|1995-03-03|2024-11-10|
+-------+---+----------+----------+
```

4. **Output File Contents**



```
24/11/10 10:03:51 WARN WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
24/11/10 10:03:51 WARN WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
+-------+---+----------+----------+
|   name|idx|       dob|start_date|
+-------+---+----------+----------+
|  Alice|  1|1990-01-01|01-01-2023|
|    Bob|  2|1985-05-15|01-02-2023|
|Charlie|  3|1992-09-09|01-03-2023|
|    Eve|  4|1987-12-12|01-04-2023|
|  Frank|  5|1993-06-22|01-05-2023|
|  Alice|  6|1990-01-01|2024-11-10|
|  David|  7|1988-08-08|2024-11-10|
|    Eve|  8|1987-12-12|2024-11-10|
|  Grace|  9|1991-10-10|2024-11-10|
|  Henry| 10|1995-03-03|2024-11-10|
+-------+---+----------+----------+
```